Martin-Luther-Universität Halle-Wittenberg
Naturwissenschaftliche Fakultät III
Studiengang Informatik

MS-Anchor: Linktexte als Ranking-Features im Zeitalter von Deep Learning

Bachelorarbeit

Maximilian Probst
geb. am: 20.01.1998 in Halle (Saale)

Matrikelnummer 217207180

1. Gutachter: Prof. Dr. Matthias Hagen
2. Gutachter: Sebastian Günther

Datum der Abgabe: 30. August 2021

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Halle (Saale), 30. August 2021

. . . . . .
Maximilian Probst

Zusammenfassung

Anchor Text wird seit Jahrzehnten von Websuchmaschinen für das Ranking von Dokumenten für Suchanfragen eingesetzt. Vor allem für traditionelle Retrieval-Modelle ist dementsprechend sehr gut untersucht, wie Anchor Text für effektive Rankings eingesetzt werden kann. Für moderne neuronale oder transformer-basierte Retrievalmodelle wurde jedoch noch nicht systematisch untersucht, ob Anchor Text immer noch Bestandteil von effektiven Rankern ist, oder einfach weggelassen werden kann. Die Grundlage für diese Arbeit stellt dabei der MS Marco-Datensatz dar, auf welchem moderne Retrievalmodelle trainiert und evaluiert werden.

Um die Nützlichkeit von Anchor Text für moderne Retrievalmodelle bewerten zu können, wird der MS Marco-Datensatz mit Anchor Texten, welche aus Crawls der Commoncrawl Foundation aus den Jahren 2016 bis 2021 generiert wurden, angereichert. Insgesamt wurden 343,05 Tebibyte an komprimierten Web Crawls nach Anchor Texten durchsucht und 4,57 Milliarden Anchor Texte extrahiert, wobei für 51,52 % der Dokumente im MS Marco Datensatz passende Anchor Texte extrahiert wurden. Anschließend wurden die extrahierten Anchor Texte auf ihre Retrieval Performance analysiert. Es zeigt sich, dass sich Anchor Texte allein nur bedingt für das Retrieval von Webseiten auf dem MS Marco Datensatz eignen, bei der Kombination mit weiteren Features konnten Anchor Texte eine bessere Retrieval-Performance als docTTTTTquery erzielen. Experimente mit 200 in dieser Arbeit erzeugten Topics zeigen, dass Anchor Texte besonders für Navigational Queries nützlich sind. Anhand dieser neuen Topics wurde eine vorangegangene Studie reproduziert, was zeigt, dass Anchor Texte auf dem MS Marco-Datensatz ein substanziell besseres Ranking produzieren, als das Seiteninhaltsverfahren und damit auch im Zeitalter von Deep Learning-Modellen noch wichtige Bestandteile für Websuche sind.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Eine Aufgabe des Information Retrieval ist es, die wesentlichen Informationen einer Webseite zu extrahieren, um diese effizient indizieren und auffinden zu können, was zu einer besseren Retrieval Performance und einer schnelleren Suche führt. Bisherige Forschung deutet darauf hin, dass Anchor Texte zu diesem Zweck herangezogen werden können [1, 15, 20], da sie zumeist sehr konkrete und kompakte Informationen enthalten und die referenzierte Webseite trotzdem gut beschreiben. Allerdings ist nicht klar, ob Anchor Texte auch für moderne, neuronale oder transformer-basierte Retrievalmodelle noch effektive Ranking-Ergebnisse liefern können, da noch keine ausführliche Forschung zu diesem Thema betrieben wurde. Aus diesem Grund wird in dieser Arbeit der MS Marco-Datensatz [26] verwendet, da er insbesondere auch für das Trainieren von Deep Learning Algorithmen entwickelt wurde.

Um Anchor Texte nutzen zu können, werden entsprechende Anchor Text-Sammlungen benötigt. Für den MS Marco-Datensatz existiert allerdings noch keine frei verfügbare Sammlung, weshalb eine eigene Anchor Text-Sammlung erstellt wurde. Viele der im MS Marco-Datensatz enthaltenen Anchor Texte verweisen jedoch auf Webseiten, welche außerhalb des Datensatzes liegen und sind deshalb ungeeignet. Aus diesem Grund werden relevante Anchor Texte stattdessen aus insgesamt sieben Crawls der Common Crawl Foundation extrahiert.

Die Erzeugung von aus dem Common Crawl abgeleiteten Datensätzen ist üblich für Forschungsfelder, die auf großen Mengen Webdaten beruhen. Mackenzie et al. [22] zum Beispiel nutzen Common Crawl Daten für die Erstellung eines 44 Mio. Dokumente umfassenden News Korpus. Brown et al. [3] nutzen die Daten, um ihr GPT-3 Language Model zu trainieren.

Kapitel 3 befasst sich mit dem Extraktionsverfahren für die Anchor Texte. Hierbei werden die Common Crawl-Daten nach sämtlichen Anchor Texten durchsucht. Die Zieladressen der Anchor Texte werden mit den Adressen der

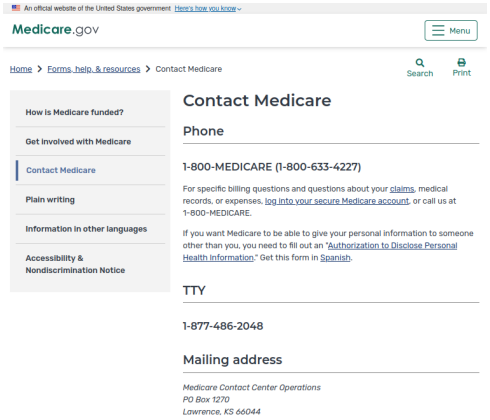

Abbildung 1.1: Die Medicare Contact Webseite (`https://www.medicare.gov/forms-help-resources/contact-medicare`).

Dokumente aus dem MS Marco-Datensatz abgeglichen, damit nur für den Datensatz relevante Anchor Texte aufgenommen werden. Durch Filterschritte wird die Qualität der Anchor Texte sichergestellt. Insgesamt wurden für diese Arbeit .. Mrd. Webseiten betrachtet und .. Mrd. Anchor Texte gesammelt.

In Kapitel 4 werden die Anchor Texte auf ihre Retrieval Performance überprüft. Hierbei werden unterschiedliche Aspekte der Anchor Texte, wie etwa das Crawl-Datum und deren Auswirkung auf die Retrieval Performance untersucht. Zusätzlich zu den Anchor Texten wurde auch der Anchor Context extrahiert und auf dessen Retrieval Performance analysiert. Es handelt sich hierbei um die Zeichen, welche den Anchor Text umgeben. Die Menge des Anchor Context wurde auf 125 Zeichen vor und 125 Zeichen nach dem Anchor Text begrenzt. Es stellt sich jedoch heraus, dass der Anchor Context in seiner reinen Form noch zu ungenau ist, um nützliche Ergebnisse zu liefern, und auch das Anchor Text-Verfahren weist keine überzeugenden Ergebnisse auf. Zusätzlich wurden Queries des ORCAS [11] Query Log für den Vergleich der Retrieval Performance zwischen Queries und Anchor Texten herangezogen. Das ORCAS Query Log wurde aus Queries der Bing Suchmaschine generiert und beinhaltet 10 Mio. Queries. Das ORCAS Query Log konnte im Vergleich zu den Anchor Texten das bessere Ranking liefern.

In Kapitel 5 wird mitunter die Nützlichkeit von Anchor Texten in der Entry Page-Suche diskutiert. Die Anchor Texte für den MS Marco-Datensatz schneiden hierbei sehr gut ab und liefern deutlich bessere Ergebnisse als das Verfahren, welches nur den Seiteninhalt nutzt. Zudem wird gezeigt, dass Anchor Texte auch für konventionelle Suchaufgaben sehr gut Ergebnisse erzielen können, wenn sie mit weiteren Features kombiniert werden.

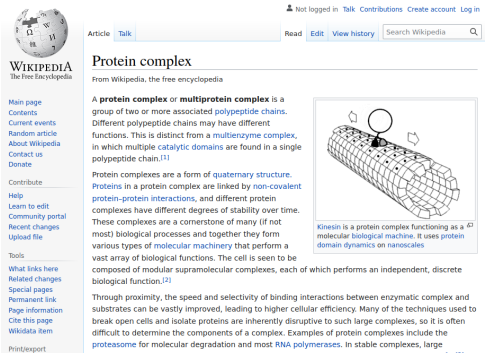

Abbildung 1.2: Der Wikipedia Artikel zu Proteinkomplexen (`https://en.wikipedia.org/wiki/Protein_complex`).

Es wurde beobachtet, dass einige Suchanfragen besser durch Anchor Texte beantwortet werden können, als durch den Seiteninhalt selbst. Zum Beispiel erzielte das Anchor Text-Verfahren für die Suchanfrage “medicare eligibility contact number” (vgl. Abb. 1.1) einen MAP@1000 Score von 1,0, während per Seiteninhalt der Score bei 0,0 verblieb.

Die Anchor Texte für die gesuchte Webseite sehen wie folgt aus: “MEDICARE Basic Information Line for sign-up/eligibility [...] Telephone Hotline Contact Medicare Medicare Contact Center [...] contact number Medicare”. Hier werden viele kontextuell korrekte Synonyme wie “number”, “telephone”, “hotline” und “information line” geliefert, ohne den Text mit irrelevanten Daten aufzublähen. Zudem ist der Kern der Anfrage (“Medicare”) deutlich öfter in den Anchor Texten vertreten, als er überhaupt im Quelldokument auftrat.

In umfangreicheren Dokumenten mit mehr Text, wie Wikipedia Artikeln, können Anchor Texte auch präzise Ergebnisse liefern. Beispielsweise die Suche nach “complex protein definition” (vgl. Abb. 1.2) ist lässt sich gut mittels Anchor Text beantworten, da der Anchor Text zum Großteil aus Wörtern wie “protein complex”, “multiprotein” und “protein complexes” besteht und das Thema der Webseite so eindeutig beschreibt. Der Artikel selbst hingegen ist sehr ausführlich, wodurch der Kerninhalt bei der Suche in den Hintergrund geraten kann. Das Anchor Text Verfahren erzielte hier wieder einen MAP@1000 von 1,0, während das Seiteninhalts-Verfahren einen Score von 0,0 erreichte.

Natürlich gilt dies nicht für alle Anchor Texte, da viele auch nur die URL der verlinkten Webseite enthalten und es nie garantiert ist, dass Anchor Texte für bestimmte Seiten existieren. Dies ist vor allem bei kleineren Seiten der Fall, welche nicht viel Präsenz im Internet aufweisen. Trotzdem zeigen diese Beispiele grundlegend, dass Anchor Texte sehr interessant für Information Retrieval sein können.

Kapitel 2

Verwandte Arbeiten

In diesem Kapitel werden Anwendungszwecke für Anchor Texte diskutiert. Zuerst werden Einsatzmöglichkeiten als Query Log-Ersatz betrachtet. Darauf folgt für das Beantworten von Navigational Queries. Aber es gibt durchaus auch andere Anwendungsfälle, welche für Anchor Texte sehr interessant sind, wie beispielsweise das automatische generieren von Snippets. Außerdem wird darauf eingegangen mit welchen Ergebnissen man bei der Arbeit mit Anchor Texten rechnen kann, und welche Korpora verwendet wurden.

Für viele Suchmaschinen bieten Anchor Texte schon seit langem wichtige Informationen für das Information Retrieval. Google [1] und ARC (Automatic Resource Compiler) [5] zählen mit zu den ersten Suchmaschinen, welche Anchor Text nutzen. Der Grund ist, dass sie präzisere Beschreibungen von Webseiten liefern können, als die Webseite selbst [1]. Außerdem können sie wichtigen Kontext für nicht-textuelle Elemente, wie beispielsweise Bilder liefern [16, 23].

Ähnlichkeit von Anchor Texten und Queries

Eiron und McCurley [15] zeigen anhand eines 2,95 Mio. Dokumente Umfassenden Korpus eines Firmen-Intranets von IBM, dass Anchor Texte und Suchanfragen sehr ähnlich zueinander sind. Sie zeigen außerdem, dass Anchor Texte zu fokussierteren und kohärenteren Ergebnissen führen können, als andere Features des Korpus. Dies hat den Grund, dass Anchor Text in der Regel weniger mehrdeutig ist, als andere Texttypen. Insbesondere ein Verfahren, welches die Titel der HTML-Dokumente nutzte, schnitt bei ihren Retrieval-Experimenten um 5 % schlechter in der Beantwortung von Queries ab, als das Anchor Text-Verfahren.

Dang und Croft [13] nutzen die Ähnlichkeit von Anchor Texten zu Suchanfragen mit dem Ziel, Query Logs durch Anchor Texte zu ersetzen, da diese im Gegensatz zu Query Logs nicht proprietären, sondern leichter zugänglich

sind. Sie verwenden Anchor Text für ein Query Substitution- und ein Query Expansion-Verfahren. Bei der Query Substitution werden einzelne Wörter der Query durch Synonyme ersetzt. Bei der Query Expansion hingegen werden Suchanfragen mit verwandten Begriffen angereichert. Sie extrahieren ihre Anchor Texte aus der TREC Gov-2 Web Collection (25 Mio. Webseiten der .gov Domäne aus dem Jahr 2004). Als Query Log nutzen Sie das MSN Query Log. Die Verfahren wurden auf drei Korpora getestet, dem WT10G Korpus, dem Robust04 Korpus, sowie dem Gov-2 Korpus. Für längere Suchanfragen ergab das Anchor Text-Verfahren auf dem WT10G Korpus sowohl mittels Query Substitution, als auch mittels Query Expansion ein besseres Ranking als das Query Log-Verfahren. Für die beiden anderen Korpora waren die Ergebnisse nahezu identisch. Das Query Log-Verfahren produzierte hier mittels Query Substitution ein leicht besseres, und mittels Query Expansion ein leicht schlechteres Ranking als das Anchor Text-Verfahren.

Craswell et al. [10] nutzen ebenfalls Anchor Texte, um Suchanfragen neu zu formulieren. Jedoch mit dem Unterschied, dass bei Dang und Croft [13] ein semimanueller Ansatz gewählt wurde, bei dem der Nutzer selbst die beste Reformulierung aus einer Liste von automatisch generierten Vorschlägen auswählt. Im Gegensatz dazu wird hier ein Verfahren vorgestellt, welches auch ohne Nutzereingaben statistisch signifikante Verbesserungen für die Retrieval Performance erreicht. Anzumerken ist, dass die verwendete Anchor Text-Sammlung deutlich umfangreicher ist, als die Korpora der zwei zuvor erwähnten Paper. Sie basiert auf dem ClueWeb09-Datensatz mit 1,04 Mrd. Webseiten und Umfasst über 1,2 Mrd. Anchor Text-Phrasen.

Anchor Texte werden von Kraft und Zien [21] für Query Refinement genutzt. Als Alternative für den Seiteninhalt bieten Anchor Texte einige Vorteile: Zum einen ist der Speicherbedarf von Anchor Text-Daten wesentlich geringer gegenüber dem ganzer Dokumentkollektionen, womit sich die Verarbeitungszeit verkürzt. Zudem wird durch die Anchor Texte die Beliebtheit der Webseiten widergespiegelt, da beliebte Webseiten häufiger von Anchor Texten referenziert werden. Dieser Faktor findet sich auch im resultierenden Query Refinement wieder und führt somit zu besseren Ergebnissen für die Retrieval Performance. Anschließend wurde auch ein Query Log für das Query Refinement genutzt. In einer User Study wurden dann alle drei Verfahren ausgewertet. Die Query Refinements, die das Query Log-Verfahren produziert, wurden zu 63 % als nützlich eingeschätzt. Die des Anchor Text-Verfahrens, wurden ebenfalls zu 63 % als nützlich eingeschätzt. Das Verfahren, welches nur den Seiteninhalt nutze, lieferte mit 51 % deutlich weniger nützlich Query Refinements.

Allerdings besteht das Problem, dass Anchor Texte im Web sehr ungleichmäßig verteilt sind, da ein Großteil aller Anchor Texte nur eine sehr kleine Anzahl an Webseiten referenziert. Vielen Webseiten können nur wenige oder gar keine

Anchor Texte zugewiesen werden. Um diese Anchor Text-Knappheit zu mildern, entwickelten Metzler et al. [24] ein Verfahren, um Anchor Texte verwandter Webseiten zu kombinieren, und somit die einzelnen Seiten mit Anchor Texten anzureichern. Um eine Webseite mit Anchor Texten anzureichern, werden zuerst alle Seiten der gleichen Domäne gesammelt, welche auf die Webseite verlinken. Dann werden sämtliche Anchor Texte gesammelt, welche eine dieser Seiten referenzieren und nicht der gleichen Domäne angehören. Die so gesammelten Anchor Texte werden nun noch gewichtet. Durch das anreichern konnte die Anzahl an URLs, für welche keine externen Anchor Texte gefunden wurden um 38 % reduziert werden. Auch die Retrieval Performance hat sich verbessert, so konnten insbesondere für längere Queries DCG-1 und DCG-5 Verbesserungen von über 10 % erreicht werden.

Dou et al. [14] untersuchen in ihrem Paper ebenfalls die Beziehungen zwischen Anchor Texten, insbesondere wenn sie von der eigenen oder einer verwandten Webseite stammen. Es werden drei Modelle diskutiert: 1. Das Baseline-Modell, welches alle Hyperlinks als unabhängig voneinander betrachtet. 2. Das Site independent Model, welches davon ausgeht, dass nur Hyperlinks von verschiedenen Seiten unabhängig voneinander sind. Dies hat die Begründung, dass Hyperlinks auf der gleichen Seite wahrscheinlich vom selben Autor verfasst worden sind und für Navigation oder Spam genutzt werden könnten. 3. Das Site Relationship Model. Es geht davon aus, dass auch Seiten miteinander in Beziehung stehen können, wie etwa Mirror Sites. Je nachdem, welches Modell genutzt wird, werden die einzelnen Anchor Texte unterschiedlich gewichtet. Ihre Experimente zeigen, dass beide Verfahren gegenüber der Baseline ein besseres Ranking produzieren.

Beantwortung von Navigational Queries mit Anchor Text

Die Suchanfragen, welche von Nutzern getätigt werden, können unterschiedlicher Natur sein. Broder [2] kategorisiert Suchanfragen in drei Klassen. Navigational, Informational und Transactional. Navigational Queries dienen dem Zweck, direkt eine bestimmte Seite zu finden. Informational Queries dagegen dienen der reinen Informationsbeschaffung. Mit Transactional Queries möchte der User Seiten erreichen, auf denen weitere Interaktionen geschehen, wie etwa Shopping oder das Downloaden bestimmter Dateien. Eine Umfrage ergab, dass 25 % der Queries Navigational Queries sind. Informational Queries werden auf einen Wert von 39 % und Transactional Queries auf 36 % geschätzt. Diese beiden Werte müssen geschätzt werden, da sie nur sehr schwer anhand von Umfragen voneinander zu trennen sind. Die folgenden Arbeiten gehen verstärkt auf den Nutzen von Anchor Texten für das Beantworten von Navigational Queries ein.

Upstill, Craswell und Hawking [28] nutzen Anchor Texte und den Seiteninhalt als Baseline für das Auffinden von Home Pages, was einen klassischen Anwendungsfall für Navigational Queries darstellt. Ziel der Arbeit war es, herauszufinden welche Art von Query-unabhängigen Daten sich am nützlichsten für das Finden von Home Pages erweisen, sowie welche Methoden nützlich sind, um bekannte und weniger bekannte Home Pages zu finden. Zusätzlich wird die Beziehung zwischen der Anzahl der eingehenden Links und dem PageRank untersucht. Die Anchor Text-Baseline schnitt deutlich besser ab als die Content-Baseline.

Auch in vielen TREC Konferenzen wurde die Effektivität von Anchor Texten untersucht. Speziell im TREC-9 Web Track [17], da im vorangegangenen Jahr keine Verbesserung der Retrieval Performance mithilfe von Hyperlinks und deren Anchor Text festgestellt werden konnte, obwohl kommerzielle Suchmaschinen dieser Zeit mit Anchor Text bereits gute Ergebnisse erzielen konnten. So haben Westerveld, Kraaij und Hiemstra [29] für diesen TREC (TREC-9) mit Anchor Texten experimentiert und die Anzahl der eingehenden Links, sowie den Linktyp untersucht. Sie wollten herausfinden, wie gut sich diese Features für die Entry Page-Suche (also ebenfalls Navigational Queries) eignen. Für den Linktyp wurde unterschieden zwischen `root`, `subroot`, `path` und `file`. `root` bezeichnet hierbei URLs, welche nur aus dem Domänennamen und optional der Endung "index.html" bestehen. Dies ist relevant, da über 70 % der URLs des Typs `root` Entry Pages sind, obwohl nur 0,6 % dieser Collection aus Entry Pages besteht. Der reine Anchor Text und Seiteninhalt erreichte in ihren Retrieval-Experimenten nahezu identische Ergebnisse, jeweils einen MRR von 0,3306 bzw. 0,3375. Danach wurden beide Features kombiniert und es konnte so ein MRR von 0,4500 erreicht werden. Schließlich wurde der Linktyp einbezogen. Er erwies sich als äußerst hilfreich und erzielte zusammen mit dem Seiteninhalt einen MRR von 0,7716. Mit allen drei Features zusammen konnte man einen MRR von 0,7745 erreichen.

Park, Ra und Jang [27] diskutieren unterschiedliche Ansätze für eine bessere Retrieval-Effektivität für den TREC-2002 web track [8]. Erläutert wird der Umgang mit dem Titel eines Dokuments, die Sentence-Query Similarity, der Umgang mit Anchor Texten und die Reranking-Phase. Die Baseline zusammen mit den Titelinformationen erreichte in Ihren Experimenten einen MRR von 0,415. Zusammen mit Anchor Texten erhöhte sich dieser Wert auf 0,445. Die Baseline zusammen mit Titelinformationen und der Sentence-Query Similarity erreichte einen MRR von 0,567. Zusammen mit Anchor Texten konnte dieses Ergebnis noch auf 0,623 erhöht werden. Anchor Texte können also auch in Verbindung mit anspruchsvolleren Methoden wie Query-Sentence Similarity zu besseren Ergebnissen beitragen.

Koolen und Kamps [20] zeigen anhand der TREC 2009 Web Track Collection [7], dass Anchor Texte auch für Ad-Hoc Suchen gute Ergebnisse erzielen können, insofern die Collection umfangreich genug ist. Diue Collection bestand hierbei sowohl aus den zu findenden Zieldokumenten, als auch aus Dokumenten, welche die Anchor Texte enthalten. Es wurde beobachtet, dass durch das Verkleinern der Collection, Anchor Texte immer schlechtere Ergebnisse erzielen, wohingegen Verfahren, welche auf dem Seiteninhalt basieren besser werden. Bei 100 % der Collection erreichte das Anchor Text-Verfahren einen MRR von 0,45, das Verfahren, welches den Seiteninhalt nutzt, erreichte nur einen Wert von 0,10. Durch Verringerung der Größe der Collection um 50 % fiel der MRR der Anchor Texte auf 0,35 und der des Seiteninhaltes stieg auf 0,45. Durch das entfernen von Dokumenten aus der Collection, nimmt die Anzahl der Anchor Texte ab, was zur Verschlechterung der Performance führt. Für Features, welche auf dem Seiteninhalt basieren, ändert sich jedoch nur, dass die Anzahl konkurierender Dokumente abnimmt, weshalb hier die Performance ansteigt.

Korpora für Training und Evaluation von Deep Learning Modellen

Im TREC 2020 Deep Learning Track [12] wird die Performance von Neural Network-Modellen, wobei zwischen Passage Ranking und Dokument Ranking unterschieden wird. Die Grundlage der Daten stellt der MS Marco [26] Datensatz dar. Er entstand aus 1 Mio. Suchanfragen aus Bings Query Log und enthält 3,3 Mio. Dokumente, sowie 8,8 Mio. Passagen aus diesen Dokumenten. Automatisch ausgewählte Passagen wurden manuell als relevant oder irrelevant für die jeweiligen Suchanfragen eingestuft.

Zusätzlich zu MS Marco wurde für den 2020 Deep Learning Track auch der ORCAS [11] Query Log bereitgestellt. Das ORCAS Query Log besteht aus 10 Mio. Bing Queries und deckt 1,4 Mio. MS Marco Dokumenten ab, welche für die Queries geklickt wurden.

Weitere Anwendungsfälle von Anchor Texten

Das Information Retrieval auf großen Web-Archiven wird durch die teils enorme Menge an Daten erschwert, da das Indexieren von kompletten Archiven zu viele Ressourcen benötigt. Aus diesem Grund nutzen Holzmann et al. [18] Anchor Texte als Hauptfeature für Ihre temporale Suchmaschine *Tempas*. Anchor Texte zu indexieren verbraucht deutlich weniger Speicher. Zudem eignen sich Anchor Texte gut, um die zentralen Webseiten einer Entität zu finden.

Chen et al. [6] argumentieren, dass Suchmaschinen auf ihren Result Pages keine Snippets, die von Zielseiten extrahiert worden anzeigen sollen, sondern verwenden statt dem Seiteninhalt Anchor Texte für die Generierung der Snippets. Diese können den Inhalt anderer Webseiten sehr kompakt und zum Teil in eigenen Worten wiedergeben. Google nutzt einen ähnlichen Ansatz, falls eine Webseite das Crawlen mittels `robots.txt` verbietet. In diesem Fall wird unter anderem Anchor Text als externe Ressource verwendet, da man so trotzdem genug Informationen über eine Seite erhalten kann, um sie zu ranken und mit einem Titel versehen zu können. [1]

Abgrenzung dieser Arbeit zu vorherigen Arbeiten

Die besprochenen Arbeiten zu Anchor Text in Verbindung mit Query Logs [10, 13, 15, 21] zeigen, dass Anchor Texte eine frei verfügbare Alternative zu den meist proprietären Query Logs darstellen können. In dieser Arbeit wird allerdings anders als in den erwähnten Papers, nicht der primäre Korpus (hier MS Marco [26]) nach Anchor Texten durchsucht. Stattdessen werden hierfür die externen Common Crawls genutzt. Dies ermöglicht, dass die Anchor Texte auch angesichts der Zeit betrachtet werden können, zu der sie aktiv waren. Außerdem ist die Menge der Anchor Text-Sammlung mit 4,57 Mrd. Anchor Texten größer, als bisherige Sammlungen. Das hier verwendete Verfahren für die Anchor Text-Extraktion ist zusätzlich skalierbar, da ohne großen Aufwand weitere Common Crawls durchsucht werden können. Dies ist nicht der Fall für Anchor Texte aus einem festgelegten Korpus.

Speziell für den MS Marco Datensatz wurden Anchor Texte allerdings zum jetzigen Zeitpunkt noch nicht im Detail behandelt, weshalb sich diese Arbeit näher mit diesem Thema befasst. Auf älteren Datensätzen haben sich Anchor Texte jedoch für Navigational Queries als sehr hilfreich erwiesen, dies soll für die Anchor Texte des MS Marco-Datensatzes untersucht werden. Zudem kann das Anlegen eines Anchor Text-Datensatzes für MS Marco durch die vielseitigen Anwendungsmöglichkeiten neue Forschung ermöglichen. Da MS Marco mit bedacht für Deep Learning Algorithmen entwickelt wurde, gilt hier ein besonderes Augenmerk im Vergleich zu Anchor Texten.

[1] `https://developers.google.com/search/docs/advanced/appearance/good-titles-snippets`

Kapitel 3

Extraktion von Anchor Texten für MS Marco

In folgendem Kapitel wird die Extraktion der Anchor Texte und deren Struktur erläutert. Anschließend wird auf die genutzten Filterschritte eingegangen, sowie ein Überblick über die gesammelten Anchor Text Daten gegeben.

3.1 Vorgehen

Als Ausgangspunkt für die Anchor Texte werden Common Crawls der Jahre 2016-2021 genutzt, pro Jahr wurde jeweils ein Crawl ausgewählt. Ein Crawl beinhaltet ca. 2-4 Mrd. Webseiten und umfasst ca. 50-80 TiB an komprimierten Daten im warc-Format.

Diese warc-Daten werden eingelesen und die resultierenden HTML Dokumente werden mithilfe der Jsoup-Bibliothek geparsed. Hierbei werden sämtliche Hyperlinks betrachtet, um die Anchor Texte zu entnehmen und entsprechende Filterschritte anzuwenden. Mehr zum Thema Filterung in Abschnitt 3.3.

Als Nächstes wird der Kontext der Anchor Texte bestimmt. Dieser Kontext beinhaltet sämtlichen Text 125 Zeichen vor und 125 Zeichen nach dem Auftreten des Anchor Textes, also insgesamt 250 Zeichen. Dies entspricht in etwa 2-3 Sätzen. Dieser Schwellwert wurde gewählt, da der Kontext im unmittelbaren Umfeld die für das Zieldokument am relevantesten Informationen enthalten sollte. Außerdem kann der Kontext bei Bedarf noch enger zugeschnitten werden.

Um die Anchor Texte präzise im Dokument identifizieren zu können, werden sie durch generierte, eindeutige Platzhalter ersetzt. Somit kann auch im selben Dokument mehrmals aufkommenden Anchor Texten zuverlässig der richtige Kontext zugewiesen werden. Um dies zu erreichen wird eine Übersetzertabelle angelegt, welche sämtliche HTML Link-Elemente und die dazugehörigen Platzhalter beinhaltet. Nach dem Einsetzen der Platzhalter wird der Kontext des

gesuchten Elements bestimmt und extrahiert. Zuletzt werden alle im extrahierten Kontext vorhandenen Platzhalter in ihre ursprüngliche Form zurück übersetzt.

3.2 Struktur extrahierter Anchor Texte

Das Schema der extrahierten Anchor Texte im JSONL-Format wird in folgender Übersicht dargestellt. JSONL wird verwendet, da es sehr flexibel ist und ohne großen aufwand erlaubt, auf Teildaten zu arbeiten.

```
AnchorElement [1]
      ↳ anchorText [2]
      ↳ anchorContext [3]
      ↳ targetUrl [4]
      ↳ targetMsMarcoDocIds[] [5]
      ↳ document [6]
            ↳ scrUrl [7]
            ↳ recordID [8]
            ↳ trecID [9]
            ↳ infoID [10]
            ↳ naughtyWords[] [11]
```

(1) `anchorElement`: Enthält alle Anchor-spezifischen Informationen eines Anchor Textes.
(2) `anchorText`: Beinhaltet den Anchor Text
(3) `anchorContext`: Beinhaltet den 250 Zeichen umfassenden Anchor Kontext.
(4) `targetUrl`: Die Ziel-URL, auf welche vom Link verwiesen wird
(5) `targetMsMarcoDocIds`: Alle MS Marco IDs, auf die dieser Anchor zeigt (i.d.R. nur eine, aber durch Ungenauigkeiten sind selten auch mehr möglich).
(6) `document`: Enthält alle Daten des Quelldokuments, aus welchem der Anchor Text stammt.
(7) `srcUrl`: URL des Quelldokuments.
(8) `redordId`: Die ID des `warcRecord`, aus dem das Dokument stammt.
(9) `trecId`: Die `TrecID` der Warc-Datei. Da es sich hier um Common Crawls handelt, welche keinem Trec zugehörig sind, dient dieses Feld als Platzhalter.
(10) `infoId`: ID des Warc Info-Eintrags (es existiert 1 Warc Info-Eintrag pro Warc-Datei) .
(11) `naughtyWords[]`: Liste von obszönen Wörtern, welche im Quelldokument vorkamen. Kann für das Filtern genutzt werden.

Diese `AnchorElement`-Objekte werden dann für jeden Crawl in einer Liste gesammelt und anschließend im JSONL-Format abgespeichert.

Im JSONL Format stellt jede Zeile ein vollwertiges JSON-Objekt dar, d.h. jede Zeile entspricht genau einem `AnchorElement`. Dadurch können die Daten leicht getrennt und zusammengeführt werden.

3.3 Filterung der Anchor Texte

Die URL Menge, welche mittels der Common Crawls betrachtet werden, beinhaltet große Mengen hier irrelevanter Daten. Da sich dieser Datensatz primär auf MS-Marco Dokumente beziehen soll, müssen entsprechend alle unwesentlichen URLs entfernt werden.

Um dies zu erreichen, wurden zunächst die 3.213.835 URLs der jeweiligen Dokumente aus dem MS-Marco Datensatz zusammen mit ihren Dokument-IDs extrahiert. Es handelt sich hierbei um 3.213.835 URLs. Danach wurden die Zieladressen der Linktexte mit den Adressen der MS-Marco Dokumente abgeglichen, falls es keine Überschneidung gibt, wird der Linktext aussortiert. Gleichzeitig werden die MS-Marco Dokument-IDs den URLs/Linktexten zugewiesen.

Die Menge der resultierenden Linktexte ist nun deutlich nützlicher, da nun sowohl die Links, als auch die Zieldokumente bekannt sind, aber sie beinhaltet immer noch unerwünschte Elemente. Weitere Filterschritte sind also vonnöten:

1. Links, welche ihre eigene Seite verlinken werden entfernt, da diese häufig nur zur Navigation genutzt werden, und diese Art von Link enthält selten informative Linktexte. Um dies zu erreichen, werden die Domänen des Quell-Links und des Ziel-Links verglichen und bei Übereinstimmung wird das Element nicht weiter betrachtet.

2. Die Stopwort-Linktexte “click”, “read”, “link”, “mail”, “here” und “open”, sowie leere Linktexte werden entfernt, da diese keinen Aufschluss über den Inhalt der Zielseite geben.

3. Zu große Linktexte, d. h. Linktexte, welche mehr als 10 Wörter beinhalten oder mehr als 60 Buchstaben zählen, werden ebenfalls entfernt.

Mit diesen Filterschritten kann die Datenmenge um die Hälfte reduziert werden, um weniger verrauschte Anchor Texte zu liefern.

3.4 Extrahierte Anchor Texte für MS Marco

Tabelle 3.1: Übersicht zum Datensatz. "Anzahl Seiten" wurde `https://commoncrawl.org/`. "Größe des Crawls" bezeichnet die komprimierte Menge Crawldaten. "Extrahierte Daten" schließt nicht nur Anchor Text-Daten, sondern auch Anchor-Kontext-Daten mit einer Kontextgröße von 250 Zeichen mit ein.

Crawl	Anzahl Seiten	Größe des Crawls	Extrahierte Daten	Anchor Texte	∅ Anchor Text pro	
					Tgt Seite	Src Seite
2016-07	1,73 Mrd.	28,57 TiB	226,75 GiB	1,05 Mrd.	331,87	3,09
2017-04	3,14 Mrd.	53,95 TiB	215,68 GiB	0,95 Mrd.	171,09	2,49
2018-13	3,20 Mrd.	67,66 TiB	187,89 GiB	0,83 Mrd.	136,07	2,10
2019-47	2,55 Mrd.	53,95 TiB	127,91 GiB	0,55 Mrd.	148,08	1,98
2020-05	3,10 Mrd.	59,94 TiB	154,51 GiB	0,67 Mrd.	159,99	1,99
2021-04	3,40 Mrd.	78,98 TiB	120,28 GiB	0,52 Mrd.	150,87	1,87

Im Laufe der Arbeit wurden Anchor Texte aus 6 Crawls extrahiert. Insgesamt wurden so 17,12 Mrd. Webseiten betrachtet und 4,57 Mrd. für MS Marco relevante Anchor Texte, samt Kontext, extrahiert. Ausgenommen sind hier sämtliche Anchor Texte, welche durch den Stopwort-Filter oder durch eine unpassende Größe heraus gefiltert wurden.

Auffällig jedoch ist der Crawl aus dem Jahr 2016. Hier wurden die wenigsten Seiten gecrawlt und trotzdem die meisten relevanten Anchor Texte gefunden. Dies hängt mit der Erstellung des Crawls zusammen, da das Verfahren des Webcrawlers von der Common Crawl Foundation im Jahr 2016 angepasst wurde. Redirects werden nun nicht sofort verfolgt, sondern werden aufgezeichnet und vom nächsten Crawl behandelt[1]. Dies wurde unter anderem Veranlasst, da die vorherigen Crawls Duplikate in den einzelnen Segmenten enthielten. Allgemein ist trotzdem mit dem Voranschreiten der Jahre, unter Berücksichtigung der Menge an betrachteten Seiten, ein fallender Trend für die Anzahl der relevanten Anchor Texte zu erkennen.

[1] `https://commoncrawl.org/2016/05/april-2016-crawl-archive-now-available/`

Kapitel 4

Anchor Text als Retrieval Feature

Um den Wert der extrahierten Linktexte im Bezug auf das Retrieval zu messen, wurden erste Retrieval Experimente mittels Anserini [30] durchgeführt. Die genutzten Anchor Texte stammen aus dem Common Crawl 2019-47.

Sampling von Anchor Texten

Zuerst müssen die Linktexte in das von Anserini genutzte jsonl Format:
`{"id"=... "content"=...}`
konvertiert werden. Hierbei stellt `id` die Dokument-ID und `content` die jeweiligen Linktexte, oder -kontexte dar. Die Linktexte bzw. -kontexte werden mittels Leerzeichen separiert.

Da die Gesamtdatenmenge recht beträchtlich ist, werden nicht alle Linktexte in das Content-Feld eingetragen, stattdessen wird eine Stichprobe genommen. Für die Strichprobe werden für jede MS-Marco Zieladresse zufällige Linktexte ausgewählt, bis ein Schwellwert erreicht wird. Als Schwellwert wurden 2000 Texte festgelegt.

Auf diese Weise kann die Datenmenge von mehreren hundert GB auf einige hundert MB reduziert werden, was sich deutlich besser für den Aufbau eines Index eignet.

Retrieval Experiment

Das Retrieval erfolgt für die 5193 Queries des TREC 2020 Deep Learning Track [12] und Ausgangspunkt der Anchor Texte ist der Common Crawl 2019-47. Es werden die folgenden 8 Varianten des BM25 Retrievalmodells verwendet:

1. BM25(Default)
2. +RM3
3. +Ax
4. +PRF
5. BM25 (Tuned)
6. +RM3 (Tuned)
7. +Ax (Tuned)
8. +PRF (Tuned)

Wobei bei "tuned" die BM25 Parameter k1 und b im Vergleich zu den standardmäßigen $k1 = 0.9$ und $b = 0.4$ auf $k1 = 3.5$ und $b = 0.95$ angepasst wurden. Diese Modelle werden dann jeweils mit MAP, Recall@100 und Recall@1000 getestet.

Ergebnisse

Tabelle 4.1: MAP, Recall@100 und Recall@1000 für den Seiteninhalt der MS Marco-Dokumente.

	Default Parameters				Tuned Parameters			
	BM25	+RM3	+Ax	+PRF	BM25	+RM3	+Ax	+PRF
MAP	0,2253	0,1597	0,1124	0,1324	0,2703	0,2231	0,1882	0,1583
R@100	0,7177	0,6626	0,5658	0,6287	0,7899	0,7772	0,7558	0,6788
R@1000	0,8783	0,8716	0,8282	0,8400	0,9241	0,9226	0,9205	0,8698

Tabelle 4.2: MAP, Recall@100 und Recall@1000 für den reinen Anchor Text des Common Crawl 2019-47 auf MS Marco-Dokumenten.

	Default Parameters				Tuned Parameters			
	BM25	+RM3	+Ax	+PRF	BM25	+RM3	+Ax	+PRF
MAP	0,0815	0,0687	0,0706	0,0560	0,0952	0,0843	0,0900	0,0673
R@100	0,2448	0,2405	0,2465	0,2303	0,2580	0,2515	0,2663	0,2351
R@1000	0,3004	0,2964	0,3023	0,2840	0,3085	0,2985	0,3096	0,2883

Tabelle 4.3: MAP, Recall@100 und Recall@1000 für den 250 Zeichen Anchor Context des Common Crawl 2019-47 auf MS Marco-Dokumenten.

	Default Parameters				Tuned Parameters			
	BM25	+RM3	+Ax	+PRF	BM25	+RM3	+Ax	+PRF
MAP	0,0385	0,0297	0,0253	0,0269	0,0634	0,0552	0,0530	0,0386
R@100	0,1589	0,1371	0,1084	0,1121	0,2309	0,2307	0,2074	0,1477
R@1000	0,2480	0,2303	0,1772	0,1735	0,3174	0,3270	0,3050	0,2205

Die reinen Anchor Texte erzielten auf dem Default BM25 Verfahren mit einem MAP wert von 0,0815 schlechtere Ergebnisse als der Seiteninhalt, welcher den Wert 0,2253 erreichte. Der Anchor Context Performte mit 0,0385 am schlechtesten. Die beiden Query Expansion-Techniken, sowie das Pseudo Relevance Feedback haben bei keinem der Verfahren zu einer Verbesserung der Ergebnisse geführt, mit der Ausnahme, dass sich mittels Axiomatic Query Expansion der Anchor Texte bei Recall@100 um 0,0017 und bei Recall@1000 um 0,0019 verbessert wurde. Die MAP hat sich jedoch für alle Verfahren, sowohl bei Anchor Text, als auch bei dem Seiteninhalt verschlechtert. Bei Anchor Text fiel die MAP durch Query Expansion um einige Prozentpunkte auf mindestens 0,0687, wärend per Seiteninhalt der MAP Wert von 0,2253 durch RM3 auf 0,1597 und durch Axiomatic Query Expansion auf 0,1124 fiel. Das Tunen der BM25 Parameter dagegen verbessert jeden Run, sowohl die des Anchor Text als auch die des Seiteninhaltes.

Die Ergebnisse der Anchor Texte mögen nicht so gut sein wie die, des konventionellen Seiteninhaltes, jedoch bestätigen sie, dass durchaus relevante Informationen aus dem Crawl extrahiert wurden. Demnach kann die Extraktion auf weiteren Crawls fortgesetzt werden.

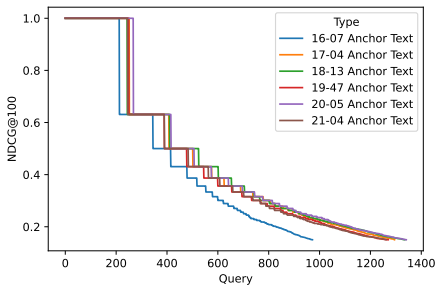

Abbildung 4.1: NDCG@100 je Query für Anchor Texte der Jahre 2016 bis 2021

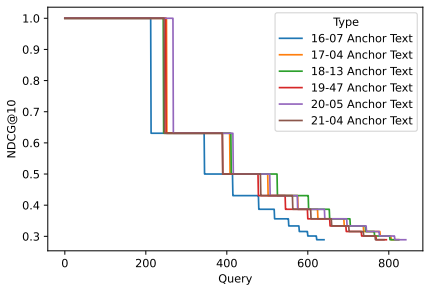

Abbildung 4.2: NDCG@10 je Query für Anchor Texte der Jahre 2016 bis 2021

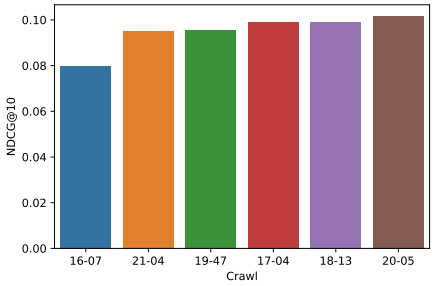

Abbildung 4.3: NDCG@10 je Crawl für Anchor Texte der Jahre 2016 bis 2021

4.1 Einfluss des Alters von Anchor Texten

Der Datensatz besteht nicht nur aus Anchor Texten zu einem bestimmten Zeitpunkt, sondern er deckt Snapshots der Jahre 2016 bis 2021 ab. Aber gibt es einen Unterschied in der retrieval Performance der einzelnen Jahre? Um dies herauszufinden wurde für jedes Jahr ein eigenes Sample erstellt. Retrieval erfolgte wieder mittels BM25 und als Performance Metriken wurden NDCG@10 und NDCG@100 gewählt.

Auffällig ist der Jahrgang 2016, welcher mit einem vergleichsweise großen Abstand negativ auffällt. An dieser Stelle sein auf Kapitel 3.4 Verwiesen, welches diese Auffälligkeit klärt. Abgesehen davon sind die Ergebnisse der restlichen Crawls recht uniform. Die Auswahl des Jahres scheint also abgesehen von Ausreißern eine eher untergeordnete Rolle zu spielen, zumindest wenn man sich in einem Zeitraum von einigen Jahren befindet.

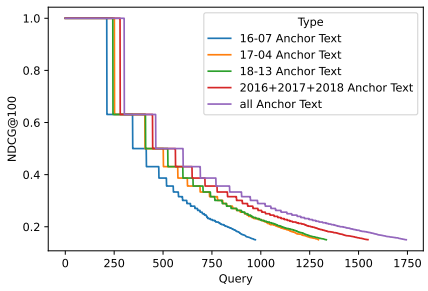

Abbildung 4.4: NDCG@100 je Query für Anchor Texte der Jahre 2016 bis 2018, sowie deren Kombination und die Kombination aller Jahre (2016–2021)

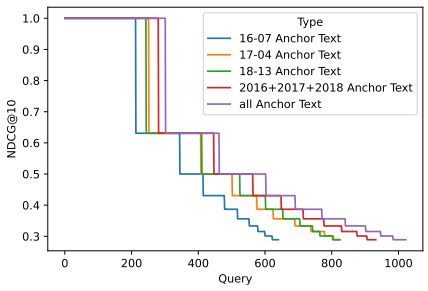

Abbildung 4.5: NDCG@10 je Query für Anchor Texte der Jahre 2016 bis 2018, sowie deren Kombination und die Kombination aller Jahre (2016–2021)

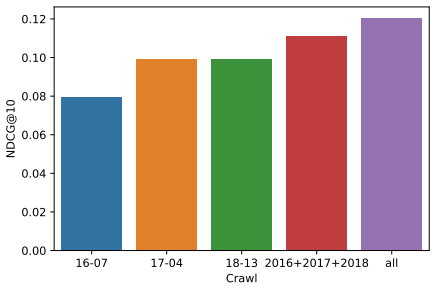

Abbildung 4.6: NDCG@10 je Jahr für Anchor Texte der Jahre 2016 bis 2018, sowie die Kombination dieser Jahre und die Kombination aller Jahre von 2016 bis 2021

Tabelle 4.4: NDCG der Anchor Texte der Jahre 2016 bis 2018, sowie 2016 bis 2021. Jeweils als Durchschnitt und als NDCG nach kombinieren der Anchor Texte

NDCG	∅(16 bis 19)	kombiniert(16 bis 18)
@10	0,0926	0,1112
@100	0,1104	0,1357

NDCG	∅(16 bis 21)	kombiniert(16 bis 21)
@10	0,0950	0,1202
@100	0,1136	0,1491

4.2 Aggregation von Anchor Texten unterschiedlichen Alters

Wenn die einzelnen Jahre also kaum Unterschiede für die Performance liefern, wie verhält sie sich dann beim Kombinieren der Jahrgänge? Schließlich sollten immer neue Seiten zu den Crawls hinzu kommen, während alte Seiten verschwinden. Demnach liegt die Vermutung nahe, dass die Vereinigung mehrerer Jahrgänge zu einer breiteren Abdeckung und demnach besseren Performance führen könnte.

Um diesen Gedanken zu überprüfen, wird die Performance der drei Jahrgänge 2016, 2017 und 2018, sowohl einzeln, als auch kombiniert, verglichen.

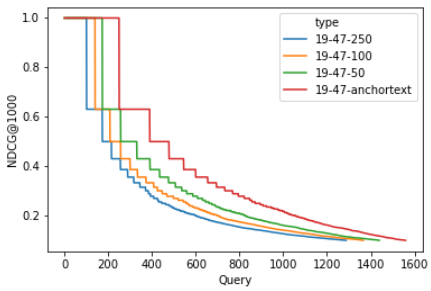

Abbildung 4.7: NDCG@1000 pro Query auf Anchor Context mit Zeichenmengen von 250, 100 und 50, sowie dem reinen Anchor Text. Basierend auf dem Crawl 19-47.

Für letzteres werden die Samples dieser drei Jahre, sowie die Samples der Jahre 2016 bis 2021 vereinigt betrachtet.

"Vereinigt" bedeutet in diesem Zusammenhang sämtliche Anchor Texte der jeweiligen Crawls in einem Sample zu kombinieren.

In Abb. 4.6 ist zu erkennen, dass das Kombinieren der drei Samples durchaus zu einer Verbesserung der Ergebnisse führt. Für NDCG@10 beläuft sich der Durchschnitt der drei einzelnen Jahre auf rund 0,0926, während die Kombination dieser drei Jahre einen Wert von rund 0,1112 erreicht. Das ist eine Verbesserung von 20,09%. Die Performance steigt sogar auf einen Wert von 0,1202, wenn die Jahre 2016 bis 2021 in Kombination genutzt werden. Das entspricht einer Verbesserung von 26,23% gegenüber dem Durchschnitt über alle sechs Jahre, welcher einen Wert von 0,0950 aufweist.

4.3 Untersuchung des Anchor Context

Im Zuge der Extraktion wurden nicht nur die reinen Anchor Texte gespeichert, sondern auch Text in deren unmittelbaren Umgebung. Dieser Anchor Context umfasst 250 Zeichen und könnte weitere Hinweise auf die Natur des Zieldokuments liefern.

Um die Relevanz der Anchor Contexte besser abzuschätzen, wurden diese mit unterschiedlichen Zeichenmengen dem Anchor Text gegenübergestellt. Hierfür wurden die Zeichenmengen 250, 100 und 50 gewählt. Wie in Abbildung 4.7 zu sehen ist, nimmt die Genauigkeit mit zunehmender Context-Größe ab. Dies muss jedoch nicht bedeuten, dass Anchor Context keinen Nutzen hat. Es muss jedoch ein Zusätzlicher Aufwand betrieben werden, um die relevanten Informationen von den irrelevanten zu trennen.

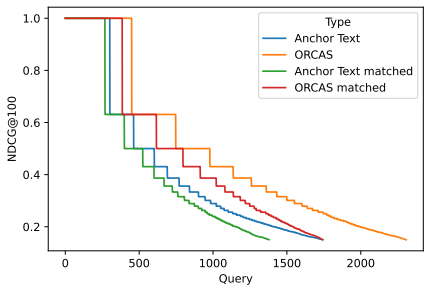

Abbildung 4.8: NDCG@100 pro Query für Anchor Texte und ORCAS Queries, sowie für auf deren Schnittmenge reduzierter Teildatensätze

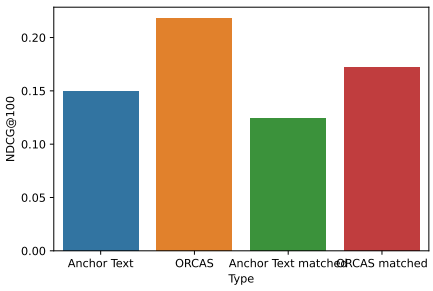

Abbildung 4.9: NDCG@100 für Anchor Texte und ORCAS Queries, sowie für auf deren Schnittmenge reduzierter Teildatensätze

4.4 Query Logs vs. Anchor Text

Der ORCAS [11] Datensatz wurde für den TREC Deep Learning Track 2020 [12] entworfen und deckt 1,4 Mio. TREC DL Dokumente mittels 10 Mio. individueller Querries ab. Der Datensatz stellt insgesamt über 18 Mio. Query-URL-Paare bereit. Das ist deutlich mehr als vergleichbare Datensätze aus vorangegangenen Jahren und befindet sich in der Größenordnung des hier untersuchten Anchor Text Datensatzes.

ORCAS wurde mithilfe von Query Logs erstellt. Dies stellt eine Besonderheit dar, da Log-Daten von Suchmaschinen in der Regel proprietärer Natur sind. ORCAS dagegen ist für Forschungszwecke frei verfügbar.

Ein direkter Vergleich zwischen dem Anchor Text und dem ORCAS Datensatz ist möglich, könnte aber aufgrund der unterschiedlichen Anzahl und Art der jeweils abgedeckten Dokumenten beeinflusst werden. Um dies entgegenzuwirken werden zusätzlich beide Datensätze auf die Schnittmenge der Dokumente reduziert. Die so entstandenen Datensätze decken jeweils 1,0 Mio. Dokumente ab.

Danach wurden die Runs wie üblich mittels Anserini und dem BM25 Verfahren erstellt.

Ergebnis

Beide ORCAS Datensätze können sich gegen die Anchor Text Datensätze durchsetzen (vgl. Abb. 4.8 und 4.9). Dieses Ergebnis ist nicht allzu überaschend, da es sich bei den ORCAS Daten um echte Queries handelt, welche eine sehr hohe Relevanz für die jeweiligen Dokumente aufweisen. Trotz allem zeigen

sich die Anchor Text-Daten auch hier wertvoll. Der große Vorteil der Anchor Texte ist hier die Verfügbarkeit, zwar ist auch ORCAS frei verfügbar, aber das ist nicht die Norm. Query Logs von Suchmaschinen sind im Normalfall nicht öffentlich zugänglich und können nicht selbst generiert werden. Umfangreiche Anchor Text-Sammlungen sind zwar auch nicht zwingend für jeden Datensatz verfügbar, allerdings haben sie den entscheidenden Vorteil, dass sie mit den entsprechenden Crawls immer extrahiert werden können. Zudem sollte man auch beachten, dass keiner der beiden Datensätze alleinstehend für das Ranking von Dokumenten ausgelegt ist, stattdessen weisen beide Datensätze einen unterstützenden Charakter auf.

Kapitel 5

Anchor Text für Navigational und Informational Queries

Nutzer von Suchmaschinen stellen Suchanfragen mit unterschiedlichen Intentionen. Dabei unterteilt Broder [2] Suchanfragen in drei unterschiedliche Arten. Informaional, Navigational und Transactional Queries. Informational Queries dienen der reinen Informationsbeschaffung. Hierbei steht im Hintergrund, von wem die gesuchten Informationen stammen. Navigational Queries haben das Ziel, eine bestimmte Webseite zu finden. Hierbei geht es weniger um das Auffinden neuer Informationen, schließlich kennt der Nutzer die Webseite bereits, da er gezielt nach ihr sucht. Transactional Queries Zielen darauf ab eine bestimmte Tätigkeit im Web zu verüben. Dies könnte das Herunterladen von Songs oder Videos sein, oder auch das Kaufen bestimmter Produkte. Der Nutzer möchte hier weder neue Informationen finden, noch sucht er eine bestimmte Entität. Diese Art von Queries wird in dieser Arbeit allerdings nicht weiter diskutiert.

Im folgenden Betrachten wir den Unterschied zwischen Navigational und informational queries anhand eines Beispiels. Ein Nutzer ist am Wetter interessiert und fragt sich, wie das Wetter den nächsten Tag sein wird. Er sucht "`wetter morgen`". Hierbei ist generell egal, welcher der vielen Wetteranbieter als Ergebnis geliefert wird, solange das Ergebnis das Wetter von Morgen beinhaltet. Es handelt sich also um eine Informational Querie. Ein anderer Nutzer möchte sich auch über das Wetter informieren, kennt aber bereits einige Wetteranbieter und entscheidet sich für AccuWeather (`https://www.accuweather.com/`). Also nutzt er die Navigational Querie "`accuweather`". Im Gegensatz zur Informational Querie können hier nur sehr wenige Seiten als relevant eingestuft werden, hauptsächlich die Home Page.

Diese Unterscheidung ist wichtig, da die unterschiedlichen Queries sehr verschiedene Anforderungen an das Retrievalmodell stellen. In einer Umfrage von Broder [2] konnte Festgestellt werden, dass es sich bei 24,5 % der gestellten

Suchanfragen um Navigational Queries, und bei ca. 39 % um Informational Queries handelt. Es handelt sich also sowohl bei Navigational, als auch bei Informational Queries um große Teile der verfassten Queries, und keiner der beiden Typen sollte vernachlässigt werden.

In dieser Arbeit wurden bislang nur Informational Queries des TREC 2020 Deep Learning Track [12] mit mäßigem Erfolg genutzt. In folgenden Abschnitten wird geprüft, wie sich die Retrieval Performance der Anchor Texte für den MS Marco-Datensatz auf Navigational Queries verhält. Außerdem wird mittels Learning-to-Rank gezeigt, wie auch auf Informational Queries gute Ergebnisse mithilfe von Anchor Texten erzielt werden können.

5.1 Navigational Queries

Im folgenden Abschnitt wird ein Retrieval Experiment zu Craswell et al. [9] nachgestellt, um die Retrieval Performance der extrahierten Anchor Texte auf dem MS Marco-Datensatz zu prüfen.

In dem Paper wird die Effektivität von Anchor Texten für das Finden von "Main Entry Points" geprüft, wobei ein Anchor Text basiertes Verfahren einem Content-basiertem gegenübergestellt wird. Das Content-Verfahren indiziert den textuellen Inhalt jedes Dokuments. Das Anchor Text-Verfahren indiziert alle Anchor Texte aus dem jeweiligen Korpus, welche das Zieldokument referenzieren.

Beide Verfahren werden mit BM25 geranked.

Evaluation

Die Evaluation erfolgt in den folgenden Schritten:

Wahl des Korpus

Es wurden zwei Korpora ausgewählt: 1. Die 100 GB und 18,5 Mio. Seiten umfassende VLC2 Kollektion, welche auf einem 1997er Internet Archiv (http://www.archive.org) von über 50 Mio Seiten basiert.

Und 2., ein 0,4 Mio. Seiten umfassender Crawl der Australian National University (ANU).

Query Paare identifizieren

Es muss ein Satz von Query Paaren generiert werden. Diese repräsentieren jeweils einen User, welcher eine Suchanfrage formuliert, um eine bestimmte Entry Page zu finden (z. B. <sigir2001, http://www.sigir2001.org/>). Hierfür wurden Dokumente ausgewählt und anschließend von Hand mit einer möglichen

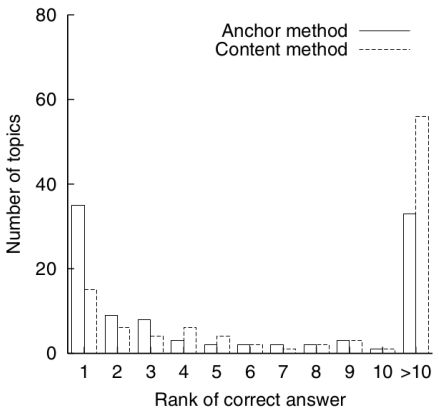

Abbildung 5.1: Die Abbildung reproduziert die Abbildung von Craswell et al. [9] für VLC2 Ergebnisse zu 100 zufällig gewählte Entry Pages (S. 255)

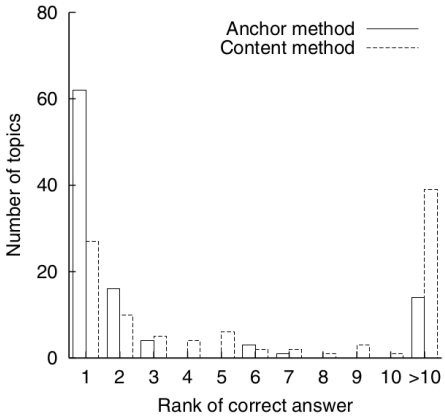

Abbildung 5.2: Die Abbildung reproduziert die Abbildung von Craswell et al. [9] für VLC2 Ergebnisse zu 100 Yahoo!-gelistete Seiten (S. 255)

Suchanfrage versehen. Für den VLC2 Korpus wurden zum einen 100 zufällige Entry Pages gewählt und zum anderen 100 zufällige durch Yahoo! verlinkte Dokumente, welche ebenfalls im VLC2 Korpus vorhanden sind. (genauer erläutern!)

Für den ANU Korpus wurde eine Seite, welche hunderte Websites der Universität verlinkt, als Grundlage für die zufällig gewählten 100 Dokumente der Query Paare genutzt. Auch hier wurden im Anschluss die Suchanfragen manuell generiert.

Verfahren laufen lassen

Für jedes der Verfahren (Anchor Text und Content basiert) die Queries über den Korpus laufen lassen.

Ergebnisse untersuchen und Effektivität messen

Hierbei müssen äquivalente Seiten beachtet werden (Seiten mit gleichem Inhalt, aber anderer URL).

Ergebnisse

Das Anchor Text Verfahren ist dem Content Verfahren in beiden Korpora überlegen (siehe Abb. 5.1 und 5.2) und erzielt in den 100 zufällig gewählten VLC2 Seiten für die erste korrekte Antwort unter den Top 10 einen MRR von

0,446. Das Content-Verfahren erreicht hier einen Wert von 0,228. Für die 100 von Yahoo! gelisteten Seiten erreicht das Anchor Text-Verfahren einen MRR Wert von 0,720, das Content Verfahren erreicht hier einen Wert von 0,370. Bei den 100 Universitätsseiten schneidet das Anchor Text-Verfahren mit einem MRR Wert von 0,790 ab, hier erreicht das Content Verfahren einen Wert von 0,321.

Beide Verfahren schneiden auf den durch Yahoo! gelisteten Seiten besser ab, als auf den zufällig gewählten Seiten.

Fazit

Die Unterschiede zwischen den Anchor Text und Content-Verfahren sind signifikant und zeigen, dass Navigational Queries sehr gut mit Anchor Texten beantwortet werden können. Nun stellt sich die Frage, ob diese Beobachtung auch mit den für diese Arbeit extrahierten Anchor Texten reproduziert werden kann.

Nachstellen des Experiments

Um das Experiment nachzustellen, müssen ebenfalls Query Paare generiert werden. Um den ersten Satz Query Paare zu generieren, werden zufällig gewählte Dokumente aus dem MS Marco Datensatz manuell mit einer entsprechenden Query versehen.

Da nicht jedes Dokument aus dem Datensatz eine Entry Page verkörpert, wird ein weiterer Filterschritt angewendet: Es werden lediglich URLs ohne Pfad betrachtet, z. B. "`http://example.com/`", nicht jedoch "`http://example.com/page123`".

Dies hat den Nachteil, dass viele Entry Pages nicht gefunden werden können (z. B. "`http://example.com/index.html`"). Westerveld, Kraaij und Hiemstra zeigen in eine Experiment, dass es sich bei 71 % der URLs, welche nur aus dem Domänenamen und optional der Endung `index.html` bestehen, um Entry Pages handelt.

Die Anzahl der verbleibenden URLs liegt nun nur noch bei 92 562, aber da nur ein Sample von 100 Queries benötigt wird, sind dies immer noch mehr als genug Dokumente. Zudem ist das Erstellen der Queries deutlich erleichtert, da es sich nun bei fast jedem Dokument um eine Entry Page handelt.

Nachdem alle 100 Query Paare generiert wurden, können die Topic und Qrel-Dateien für das Retrieval mit Anserini angelegt werden. Die Topics-Datei besteht aus den Spalten `topic_id` und `query`, letzteres wird mit den 100 erstellten Queries gefüllt. Jedem Query wird seine eigene `topic_id` zugewiesen.

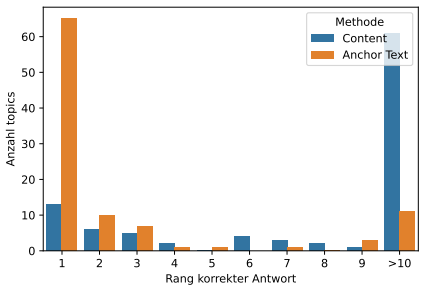

Abbildung 5.3: Ergebnisse für 100 zufällig ausgewählte Entry Pages auf MS Marco

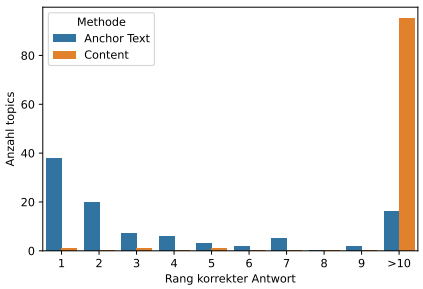

Abbildung 5.4: Ergebnisse für 100 zufällig ausgewählte Entry Pages mit Top 500 Domains auf MS Marco

Die Qrels-Datei besteht aus den Spalten `topic_id`, `iter`, `id` und `relevancy`. Hier wird die Query über die vorhin eingeführte `topic_id` mit der gesuchten DokumentID verknüpft. Die `iter` Spalte dient lediglich als Platzhalter und beinhaltet nur Nullen. Die `relevancy` Spalte beinhaltet nur Einsen, da jedes aufgeführte Dokument für seinen jeweiligen Topic/Query relevant ist.

Mithilfe der Topics und Qrels kann nun mit dem Retrieval begonnen werden. Ähnlich wie in der Vorlage wird auch hier das BM25 Verfahren genutzt.

Schlussendlich werden die erstellten Runs noch ausgewertet und man kommt zu den Ergebnissen auf Abb. 5.3.

Um die Relevanz der durch Yahoo! gelisteten Seiten zu emulieren wurden Seiten gewählt, deren Domain laut der Alexa Internet Inc. unter den Top 500 geranked wurden. Dies hatte andere Konsequenzen als zuerst vermutet, aber dazu später mehr. Übrig blieben 151 Webseiten, woraus 100 zufällig für den Run gewählt wurden (siehe Abb. 5.4).

Die Ergebnisse des Runs aus zufällig gewählten Entry Pages sind tatsächlich sehr ähnlich zu denen aus Craswell, Hawking und Robertsons Arbeit [9]. Anchor Texte zeigen sich also auch für den MS Marco-Datensatz als sehr hilfreich im Umgang mit Entry Pages.

Interessant ist, dass die Ergebnisse der zufälligen Entry Pages auf MS Marco etwas näher an den Ergebnissen der ANU Collection bzw. der VLC2 Entry Pages, welche bereits durch Yahoo! behandelt wurden liegen, anstatt der komplett zufälligen VLC2 Entry Pages. Dies könnte unter anderem die Ursache haben, dass MS Marco bereits auf Click Logs beruht und somit bringen diese Seiten bereits eine gewisse Relevanz mit sich mit, ähnlich wie die Yahoo! Strategie.

Viel spannender jedoch ist das Ergebnis für die beliebten Domains, welche das Pendant zu den in Yahoo! gelisteten Seiten darstellen sollte. Die Anchor Text-Variante schnitt etwas schlechter als bei den komplett zufällig gewählten

Entry Pages ab, liefert aber immer noch sehr gute Ergebnisse. Die Content-Variante hingegen hatte sehr starke Probleme mit den beliebten Domains und hat nur wenige richtige Seiten mit einem Rang von weniger als 10 eingestuft.

Nachdem sichergestellt wurde, dass es sich nicht um einen Fehler in der Evaluierung handelt, wurde diese überraschende Diskrepanz genauer untersucht. Dafür wurde die Suchanfrage "accuweather", welche zu dessen Homepage "`https://www.accuweather.com/`" führen soll als Beispiel gewählt.

Für diese Seite fällt auf, dass der Begriff "accuweather" weder im `title` noch im `content`-Feld auftaucht. Dagegen tritt der Begriff viel häufiger auf Seiten auf, welche über AccuWeather, oder allgemein über Wetterapps berichten, z. B. "`http://www.wikihow.com/Get-AccuWeather-for-Windows-8`" und "`https://www.macworld.com/article/2875913/the-best-weather-apps-for-ios.html`". Diese Art von Seiten ist allerdings komplett irrelevant für Navigational Queries.

Die mithilfe von Anchor Texten ausgesuchten Seiten weisen dagegen andere Eigenschaften auf, weshalb sie sich deutlich besser für navigational Queries eignen. So wird hier neben der Homepage auch die AccuWeather Seite im Play Store und im App Store verlinkt. Häufig werden auch etwas zu spezifische Seiten wie "`https://www.accuweather.com/en/europe-weather`" ermittelt, allerdings stimmt hier zumindest die Domain, weshalb der Eintrag für navigational Queries noch interessant ist.

Wenn man nun die Frage stellt, weshalb das Content-Verfahren so schlecht auf den bekanntent Seiten funktioniert, so kann dies folgendermaßen begründet werden: Wenn eine Seite sehr bekannt ist, dann wird es mit Sicherheit auch mehr Quellen geben, welche über diese Seite berichten und Reviews oder Vergleiche etc. schreiben. Das führt dazu, dass es viele Seiten mit den jeweiligen Schlagwörtern gibt. Hinzu kommt, dass die gesuchten Seiten ihren eigenen Namen oft nur wenige Male auf ihrer Homepage erwähnen. Anchor Texte hingegen würden beispielsweise weniger auf Reviewseiten verweisen, sondern eher von letzterem auf die entsprechende Homepage.

Zusammenfassend kann man sagen, dass das Content-Verfahren eher Seiten liefert, welche über das gewählte Thema berichten, wohingegen das Anchor Text-Verfahren eher Seiten liefert, welche mit der gesuchten Entry Page verwandt sind. Demzufolge eignet sich der Content kaum für die Entry Page-Suche.

5.2 Kombination von Anchor Texten mit anderen Features

Gerne werden Anchor Texte mit unterschiedlichen Features kombiniert. Dabei fällt auf, dass Anchor Texte auch in Verbindung mit bereits guten Retrieval-

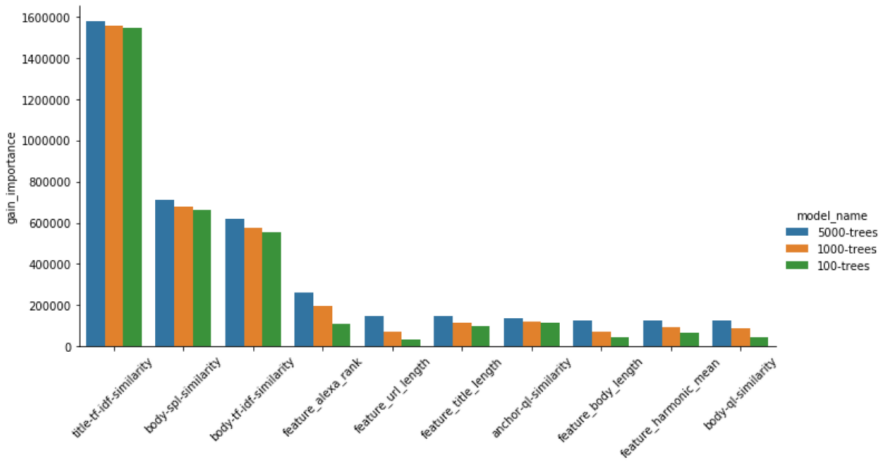

Abbildung 5.5: Feature-Importance Plot der 10 wichtigsten Features des Learning-to-Rank-Verfahrens. Anzumerken ist, dass Anchor Text (`anchor-ql-similarity`) das siebtwichtigste Feature ist.

modellen wie bei Westerveld, Kraaij und Hiemstra [29], oder bei komplexeren Verfahren wie bei Park, Ra und Jang [27], zu Verbesserungen des Gesamtergebnisses beitragen können. Die Features und Gewichtungen wurden hierbei manuell festgelegt. In dieser Arbeit wird auf diesem Ansatz aufgebaut, allerdings werden die Features hier nicht manuell festgelegt und gewichtet, sondern in einem feature-basierten Learning-to-Rank-Verfahren automatisch mithilfe der MS Marco Trainingsdaten bestimmt. Das trainierte Modell vereinigt die genutzten Features in einem Ranking-Modell. Als Learning-To-Rank Verfahren wird das State-of-the-Art Verfahren LambdaMART [4] in der Implementierung von LightGBM [19] verwendet.

Das LambdaMART-Modell wurde mit 5000 Bäumen und 50 Features auf den MS Marco Training Queries trainiert. Anchor Texte sind in dem Verfahren das siebtwichtigste Feature (vgl. `anchor-ql-similarity` Abb. 5.5). Das wichtigste Feature ist die title-df-idf-similarity. Das zweit- und drittwichtigste Feature, jedoch mit etwas Abstand, sind jeweils die body-spl-similarity und die body-tf-idf-similarity. Insgesamt kommen Titel-Features zwei mal und Body-Features 4 mal unter den Top 10 vor.

Um die Retrieval Performance zu beurteilen, wurde das Verfahren auf MRR und NDCG@10 geprüft, und mit Anserinis Baseline[1], sowie dem docTTTTTquery-Verfahren[2] verglichen (vgl. Tabelle 5.1). Mit seinem MRR Wert von 0,9444

[1] `https://github.com/castorini/anserini/blob/master/docs/regressions-dl20-doc.md`

[2] `https://github.com/castorini/anserini/blob/master/docs/regressions-dl20-doc-docTTTTTquery-per-doc.md`

Tabelle 5.1: Learnin-to-Rank-Verfahren im Vergleich mit der Deep Learning 2020 und docTTTTTquery Baseline. Default BM25 Parameter: k1=0,9 b=0,4. Getunte BM25 Parameter für die Anserini Baseline: k1=3,44 b=0,87. Getunte BM25 Parameter für doc5Tquery: k1=4,68 b=0,87.

		MRR	NDCG@10
Learning-to-Rank	LambdaMART (5000 trees)	0,9444	0,5957
Anserini Baseline	BM25	0,8521	0,5271
Anserini Baseline	BM25 (tuned)	0,8641	0,5087
doc5Tquery	BM25	0,9369	0,5885
doc5Tquery	BM25 (tuned)	0,9439	0,5852

konnte es sowohl die Anserini Baseline, als auch das docTTTTTquery-Verfahren selbst gegen getunte Parameter schlagen. Gleiches gilt für den NDCG@10 Wert von 0,5957. Anchor Texte können also auch für Informational Queries nützliche Hilfe leisten, solange sie mit anderen Ranking Features kombiniert werden, die sich ergänzen.

Kapitel 6

Fazit

Im Rahmen dieser Arbeit wurde der MS Marco Datensatz mit 1,57 Mrd. Anchor Texten angereichert, um zu evaluieren, ob Anchor Text auch für Deep Learning Modelle, welche üblicherweise auf dem MS Marco Datensatz trainiert werden, noch ein wichtiges Retrieval Feature darstellt. Die extrahierten Anchor Texte umfassen komprimiert 1,01 Tebibyte und decken 51,52 % (insgesamt 1,70 Millionen) Dokumente des MS Marco Datensatzes ab.

Im Rahmen dieser Arbeit wurde ein komprimiert 1,01 TiB umfassender Datensatz generiert, welcher 4,57 Mrd. Anchor Texte inklusive deren Kontext für 1,70 Mio. Webseiten des MS Marco-Datensatzes enthält. Anhand dieser Anchor Texte wurde die Retrieval Performance auf dem MS Marco-Datensatz gemessen.

Unsere Experimente auf Informational Queries für MS Marco zeigen, dass auch der Anchor Text alleine, ohne Einsatz des eigentlichen Dokumentinhaltes informational queries beantworten kann. Dabei liefert die alleinige Nutzung von Anchor-Texten einen MAP von 0,0815, verglichen mit einem MRR von 0,2253 den der Content erhält. Wenn Anchor Texte mit Content und weiteren Features in einer feature-basierten Learning-To-Rank Pipeline kombiniert werden, schlägt dies die Effektivität von dem modernen Retrievalmodell docTTTTTquery welches nur auf dem Inhalt arbeitet.

Für Navigational Queries erzielten Anchor Texte weitaus bessere Ergebnisse als der Seiteninhalt, insbesondere wenn die gesuchten Webseiten beliebter Domänen angehören. Es wurde auch gezeigt, dass Anchor Texte durchaus hilfreich für Informational Queries sind, vorausgesetzt man kombiniert sie mit weiteren Features. Hierfür wurde ein Learnig-to-Rank Verfahren genutzt, welches mit modernen Retrievalmodellen mithalten kann.

Die in dieser Arbeit erzeugten Anchor Text-Sammlungen erlauben neue, auf vorherigen Datensätzen unmögliche Auswertungen, da die in dieser Arbeit extrahierten Anchor Texte sowohl die Auswertung zeitlicher Aspekte erlauben,

da Anchor Texte im Zeitraum von 2015 bis 2021 extrahiert wurden, als auch einem viel größeren Umfang aufweist, als vorherige Anchor Text Sammlungen. Zukünftige Arbeit kann zum Beispiel die Ähnlichkeit von Anchor Texten und Queries, die Im ORCAS [11] Query Log veröffentlicht wurden untersuchen, und dabei existierende Studien wie die von Eiron und McCurley [15] erweitern.

Die hier bereitgestellten Anchor Texte können auch für Anwendungsfälle abseits des Information Retrieval genutzt werden. Das Generieren neuer Snippets für Suchmaschinen [6] ist einer davon. Mithilfe von Anchor Texten kann hier auf den Zugriff des Seiteninhaltes verzichtet werden.

Twitter ist ebenfalls sehr interessant für die Gewinnung von Anchor Texten, da sehr viele Tweets andere Webseiten verweisen und sich mit wenigen Sätzen über sie äußern. Da Tweets in einem anderen Kontext und mit der Intention der Kommunikation geschrieben werden, könnten sie nützlicher für Informational Queries sein, als normale Anchor Texte. So könnte man beispielsweise den gesamten Tweet als Anchor Text, bzw. Anchor Context sehen [25]. MS Marco verfügt ebenfalls über Twitter Links. Allerdings handelt es sich hierbei größtenteils um ganze Profile und nicht um einzelne Tweets, weshalb sich andere Korpora ggf. besser hierfür eignen.

Literaturverzeichnis

[1] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Networks*, 30(1-7):107–117, 1998. doi: 10.1016/S0169-7552(98)00110-X. URL https://doi.org/10.1016/S0169-7552(98)00110-X.

[2] Andrei Z. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002. doi: 10.1145/792550.792552. URL https://doi.org/10.1145/792550.792552.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

[4] Chris J. C. Burges, Krysta M. Svore, Qiang Wu, and Jianfeng Gao. Ranking, boosting, and model adaptation. Technical Report MSR-TR-2008-109, October 2008. URL https://www.microsoft.com/en-us/research/publication/ranking-boosting-and-model-adaptation/.

[5] Soumen Chakrabarti, Byron Dom, Prabhakar Raghavan, Sridhar Rajagopalan, David Gibson, and Jon M. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. *Comput.*

Networks, 30(1-7):65–74, 1998. doi: 10.1016/S0169-7552(98)00087-7. URL https://doi.org/10.1016/S0169-7552(98)00087-7.

[6] Wei-Fan Chen, Shahbaz Syed, Benno Stein, Matthias Hagen, and Martin Potthast. Abstractive snippet generation. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, editors, *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 1309–1319. ACM / IW3C2, 2020. doi: 10.1145/3366423.3380206. URL https://doi.org/10.1145/3366423.3380206.

[7] Charles L. A. Clarke, Nick Craswell, and Ian Soboroff. Overview of the TREC 2009 web track. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009, Gaithersburg, Maryland, USA, November 17-20, 2009*, volume 500-278 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2009. URL http://trec.nist.gov/pubs/trec18/papers/WEB09.OVERVIEW.pdf.

[8] Nick Craswell and David Hawking. Overview of the TREC-2002 web track. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of The Eleventh Text REtrieval Conference, TREC 2002, Gaithersburg, Maryland, USA, November 19-22, 2002*, volume 500-251 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2002. URL http://trec.nist.gov/pubs/trec11/papers/WEB.OVER.pdf.

[9] Nick Craswell, David Hawking, and Stephen E. Robertson. Effective site finding using link anchor information. In W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors, *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA*, pages 250–257. ACM, 2001. doi: 10.1145/383952.383999. URL https://doi.org/10.1145/383952.383999.

[10] Nick Craswell, Bodo Billerbeck, Dennis Fetterly, and Marc Najork. Robust query rewriting using anchor data. In Stefano Leonardi, Alessandro Panconesi, Paolo Ferragina, and Aristides Gionis, editors, *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*, pages 335–344. ACM, 2013. doi: 10.1145/2433396.2433440. URL https://doi.org/10.1145/2433396.2433440.

[11] Nick Craswell, Daniel Campos, Bhaskar Mitra, Emine Yilmaz, and Bodo Billerbeck. ORCAS: 18 million clicked query-document pairs for analyzing

search. *CoRR*, abs/2006.05324, 2020. URL `https://arxiv.org/abs/2006.05324`.

[12] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. Overview of the TREC 2020 deep learning track. *CoRR*, abs/2102.07662, 2021. URL `https://arxiv.org/abs/2102.07662`.

[13] Van Dang and W. Bruce Croft. Query reformulation using anchor text. In Brian D. Davison, Torsten Suel, Nick Craswell, and Bing Liu, editors, *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*, pages 41–50. ACM, 2010. doi: 10.1145/1718487.1718493. URL `https://doi.org/10.1145/1718487.1718493`.

[14] Zhicheng Dou, Ruihua Song, Jian-Yun Nie, and Ji-Rong Wen. Using anchor texts with their hyperlink structure for web search. In James Allan, Javed A. Aslam, Mark Sanderson, ChengXiang Zhai, and Justin Zobel, editors, *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 227–234. ACM, 2009. doi: 10.1145/1571941.1571982. URL `https://doi.org/10.1145/1571941.1571982`.

[15] Nadav Eiron and Kevin S. McCurley. Analysis of anchor text for web search. In Charles L. A. Clarke, Gordon V. Cormack, Jamie Callan, David Hawking, and Alan F. Smeaton, editors, *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 28 - August 1, 2003, Toronto, Canada*, pages 459–460. ACM, 2003. doi: 10.1145/860435.860550. URL `https://doi.org/10.1145/860435.860550`.

[16] V. Harmandas, Mark Sanderson, and Mark D. Dunlop. Image retrieval by hypertext links. In Nicholas J. Belkin, A. Desai Narasimhalu, Peter Willett, William R. Hersh, Fazli Can, and Ellen M. Voorhees, editors, *SIGIR '97: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 27-31, 1997, Philadelphia, PA, USA*, pages 296–303. ACM, 1997. doi: 10.1145/258525.258594. URL `https://doi.org/10.1145/258525.258594`.

[17] David Hawking. Overview of the TREC-9 web track. In Ellen M. Voorhees and Donna K. Harman, editors, *Proceedings of The Ninth Text REtrieval Conference, TREC 2000, Gaithersburg, Maryland, USA, November 13-16, 2000*, volume 500-249 of *NIST Special Publication*. National Institute of

Standards and Technology (NIST), 2000. URL http://trec.nist.gov/pubs/trec9/papers/web9.pdf.

[18] Helge Holzmann, Wolfgang Nejdl, and Avishek Anand. Exploring web archives through temporal anchor texts. In Peter Fox, Deborah L. McGuinness, Lindsay Poirier, Paolo Boldi, and Katharina Kinder-Kurlanda, editors, *Proceedings of the 2017 ACM on Web Science Conference, WebSci 2017, Troy, NY, USA, June 25 - 28, 2017*, pages 289–298. ACM, 2017. doi: 10.1145/3091478.3091500. URL https://doi.org/10.1145/3091478.3091500.

[19] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3146–3154, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

[20] Marijn Koolen and Jaap Kamps. The importance of anchor text for ad hoc search revisited. In Fabio Crestani, Stéphane Marchand-Maillet, Hsin-Hsi Chen, Efthimis N. Efthimiadis, and Jacques Savoy, editors, *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 122–129. ACM, 2010. doi: 10.1145/1835449.1835472. URL https://doi.org/10.1145/1835449.1835472.

[21] Reiner Kraft and Jason Y. Zien. Mining anchor text for query refinement. In Stuart I. Feldman, Mike Uretsky, Marc Najork, and Craig E. Wills, editors, *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004*, pages 666–674. ACM, 2004. doi: 10.1145/988672.988763. URL https://doi.org/10.1145/988672.988763.

[22] Joel M. Mackenzie, Rodger Benham, Matthias Petri, Johanne R. Trippas, J. Shane Culpepper, and Alistair Moffat. Cc-news-en: A large english news corpus. In Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux, editors, *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 3077–3084. ACM,

2020. doi: 10.1145/3340531.3412762. URL https://doi.org/10.1145/3340531.3412762.

[23] Oliver A McBryan. Genvl and wwww: Tools for taming the web. In *Proceedings of the first international world wide web conference*, volume 341. Citeseer, 1994.

[24] Donald Metzler, Jasmine Novak, Hang Cui, and Srihari Reddy. Building enriched document representations using aggregated anchor text. In James Allan, Javed A. Aslam, Mark Sanderson, ChengXiang Zhai, and Justin Zobel, editors, *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 219–226. ACM, 2009. doi: 10.1145/1571941.1571981. URL https://doi.org/10.1145/1571941.1571981.

[25] Gilad Mishne and Jimmy J. Lin. Twanchor text: a preliminary study of the value of tweets as anchor text. In William R. Hersh, Jamie Callan, Yoelle Maarek, and Mark Sanderson, editors, *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pages 1159–1160. ACM, 2012. doi: 10.1145/2348283.2348518. URL https://doi.org/10.1145/2348283.2348518.

[26] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268, 2016. URL http://arxiv.org/abs/1611.09268.

[27] Eui-Kyu Park, Dong-Yul Ra, and Myung-Gil Jang. Techniques for improving web retrieval effectiveness. *Inf. Process. Manag.*, 41(5):1207–1223, 2005. doi: 10.1016/j.ipm.2004.08.002. URL https://doi.org/10.1016/j.ipm.2004.08.002.

[28] Trystan Upstill, Nick Craswell, and David Hawking. Query-independent evidence in home page finding. *ACM Trans. Inf. Syst.*, 21(3):286–313, 2003. doi: 10.1145/858476.858479. URL https://doi.org/10.1145/858476.858479.

[29] Thijs Westerveld, Wessel Kraaij, and Djoerd Hiemstra. Retrieving web pages using content, links, urls and anchors. In Ellen M. Voorhees and Donna K. Harman, editors, *Proceedings of The Tenth Text REtrieval Conference, TREC 2001, Gaithersburg, Maryland, USA, November 13-16,*

2001, volume 500-250 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2001. URL `http://trec.nist.gov/pubs/trec10/papers/TNO-UTwente-trec10-final.pdf`.

[30] Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the use of lucene for information retrieval research. In Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White, editors, *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 1253–1256. ACM, 2017. doi: 10.1145/3077136.3080721. URL `https://doi.org/10.1145/3077136.3080721`.